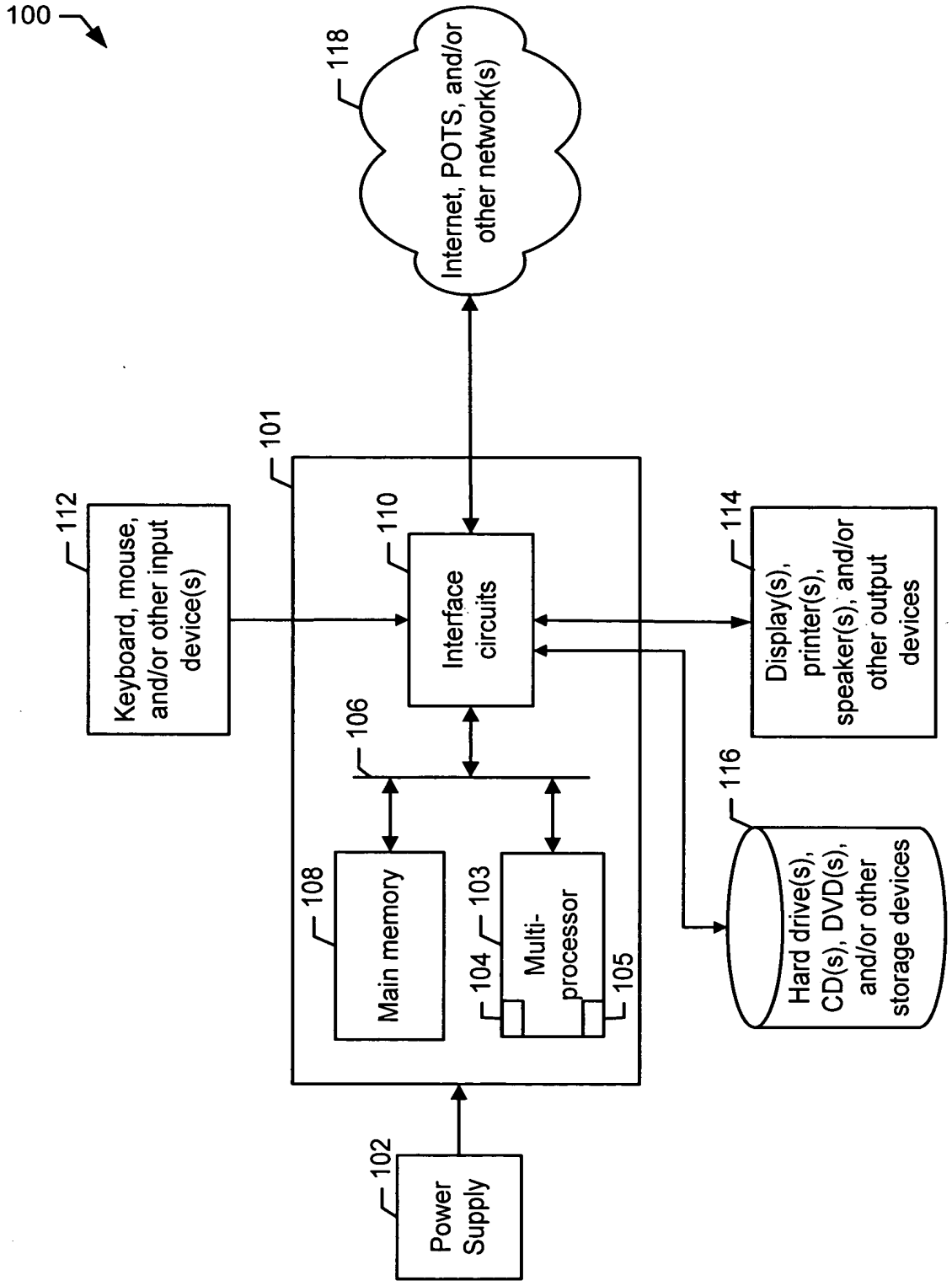

100
118
Internet, POTS, and/or other network(s)
101
112
Keyboard, mouse, and/or other input device(s)
110
Interface circuits
114
Display(s), printer(s), speaker(s), and/or other output devices
106
116
Hard drive(s), CD(s), DVD(s), and/or other storage devices
108
Main memory
103
Multi-processor
104
105
102
Power Supply

FIG. 1

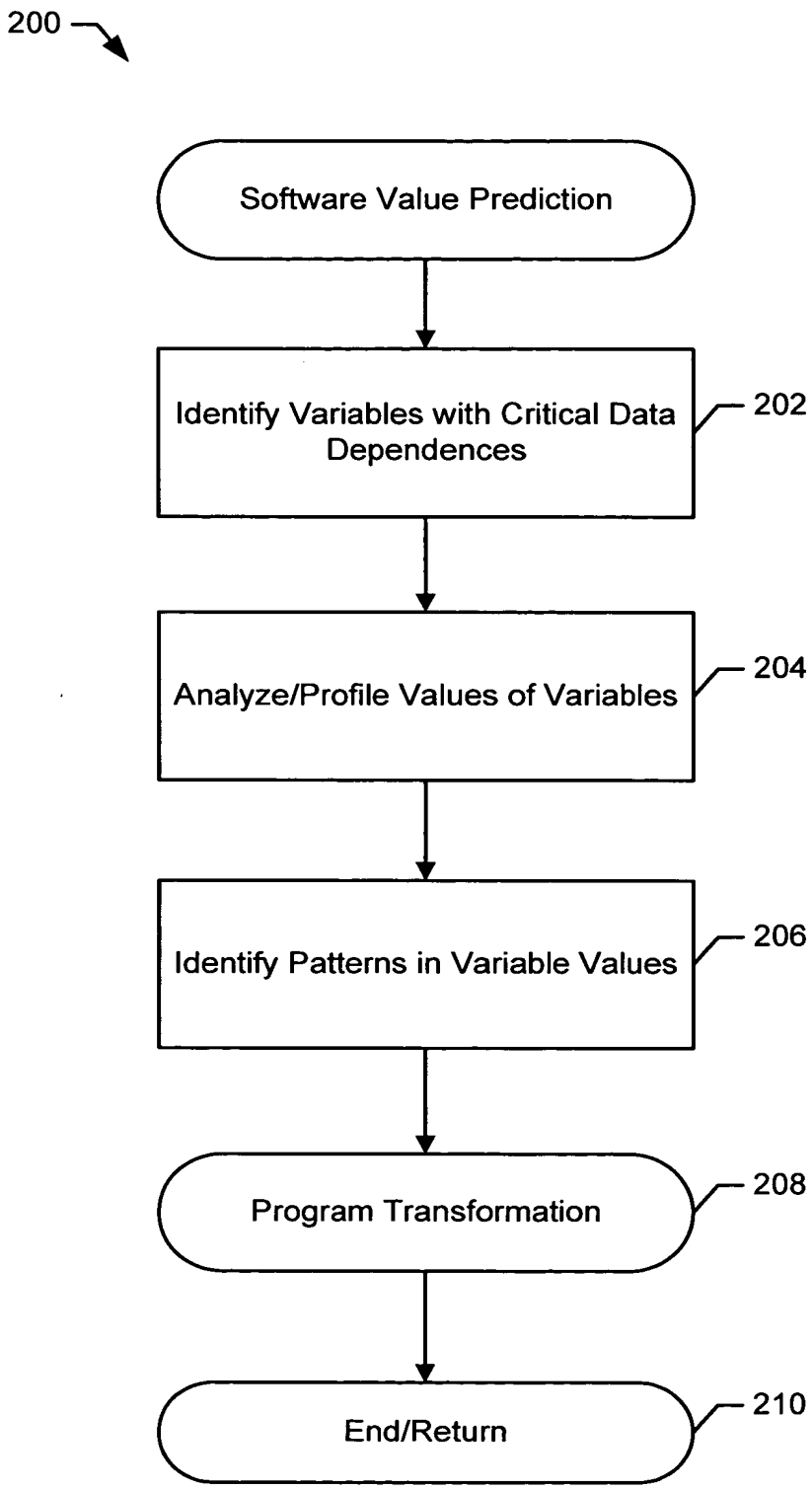

200
Software Value Prediction
Identify Variables with Critical Data Dependences
202
Analyze/Profile Values of Variables
204
Identify Patterns in Variable Values
206
Program Transformation
208
End/Return
210

FIG. 2

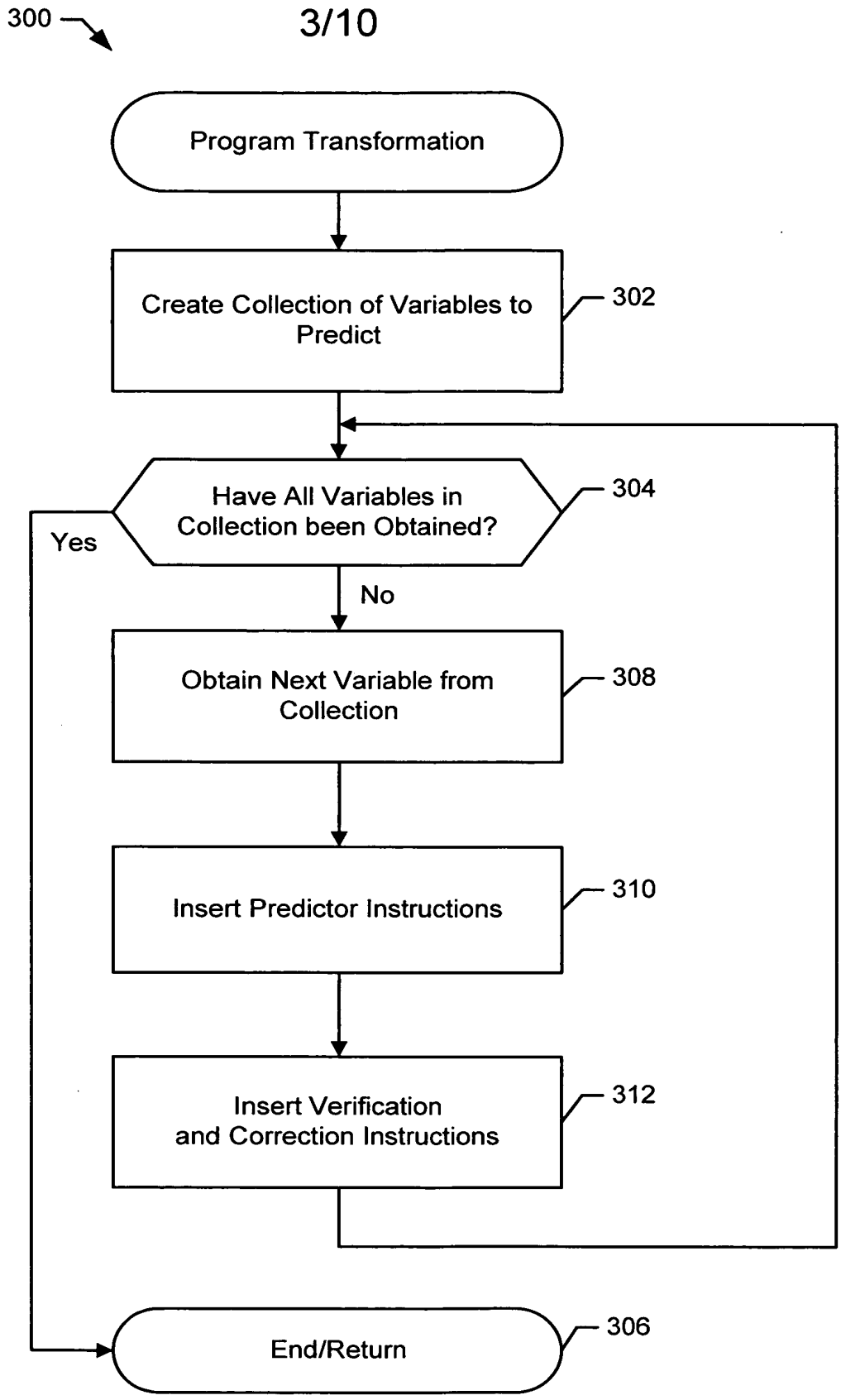

300
Program Transformation
Create Collection of Variables to Predict
302
Have All Variables in Collection been Obtained?
304
Yes
No
Obtain Next Variable from Collection
308
Insert Predictor Instructions
310
Insert Verification and Correction Instructions
312
End/Return
306

FIG. 3

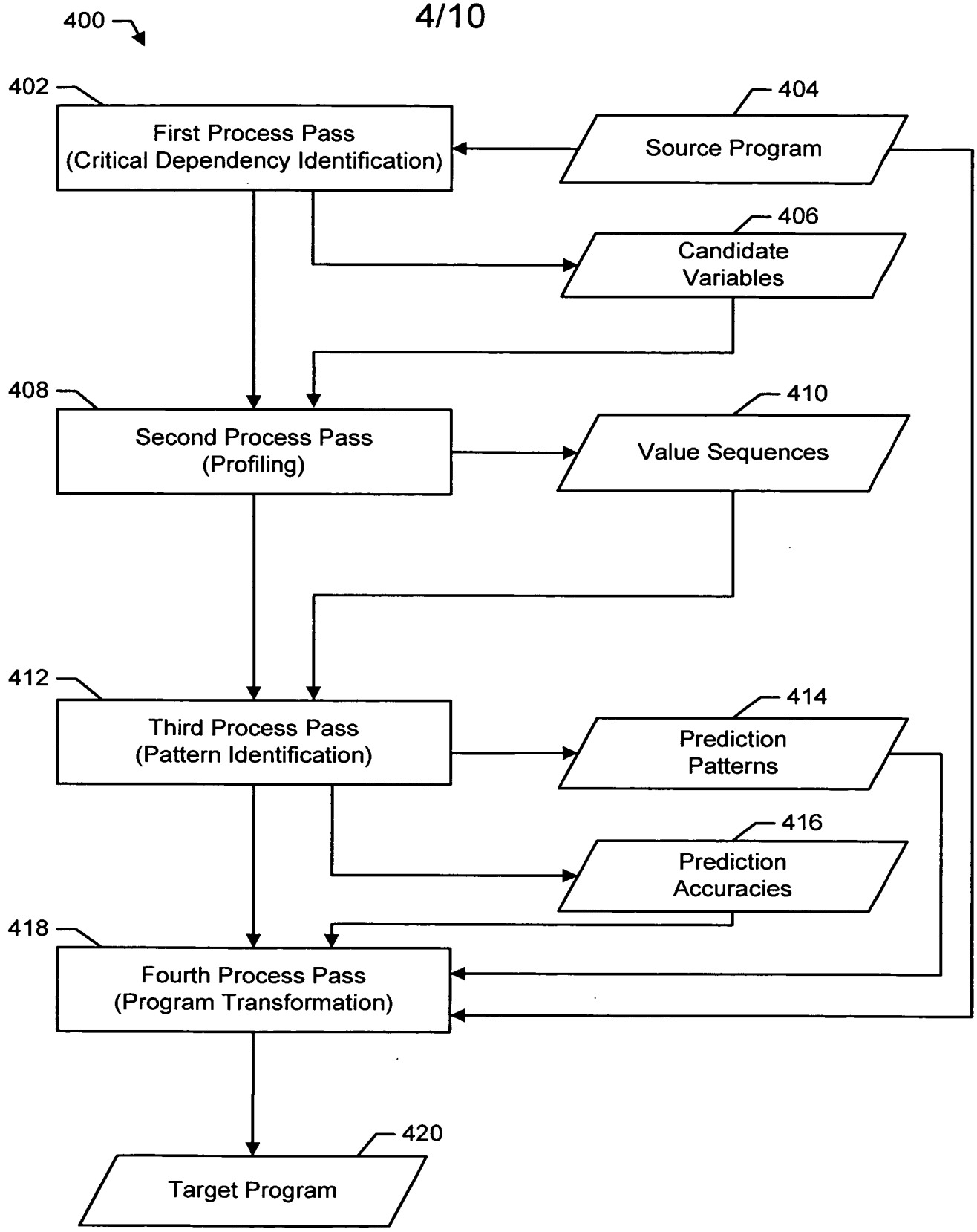
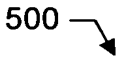
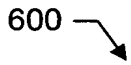

400
402
First Process Pass
(Critical Dependency Identification)
404
Source Program
406
Candidate
Variables
408
Second Process Pass
(Profiling)
410
Value Sequences
412
Third Process Pass
(Pattern Identification)
414
Prediction
Patterns
416
Prediction
Accuracies
418
Fourth Process Pass
(Program Transformation)
420
Target Program

FIG. 4



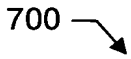

```
502:    foo( x );
504:    S1;
506:    x = bar();
508:    S2;
510:    S3;
512:    tar( x );
514:    S4;
```

FIG. 5




```
602:    pred_x = pred( );
604:    tar( pred_x );
502:    foo( x );
504:    S1;
506:    x = bar();
508:    S2;
510:    S3;
606:    if( x != pred_x )
608:        recover_tar( x );
514:    S4;
```

FIG. 6




```
702:    L:  S1;
704:        fork( L );
706:        S2;
708:        foo( x );
710:        S3;
712:        x = bar( );
714:        S4;
716:        if (cont)  goto L;
```

FIG. 7

800

```
802:          pred_x = x;
702:    L:    S1;
804:          x = pred_x;
806:          pred_x = pred( );
704:          fork( L );
706:          S2;
708:          foo( x );
710:          S3;
712:          x = bar( );
714:          S4;
808:          if (x != pred_x)
810:                pred_x = x;
716:          if (cont)  goto L;
```

FIG. 8

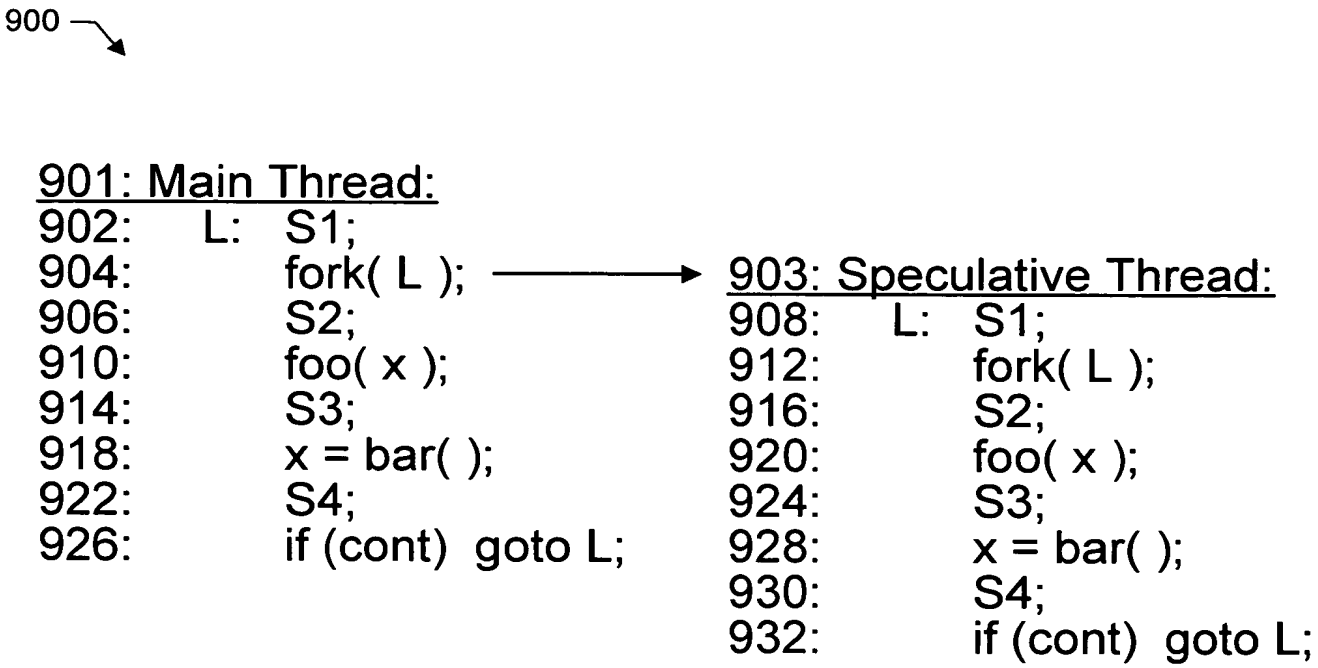

900
901: Main Thread:
902: L: S1;
904: fork(L);
906: S2;
910: foo(x);
914: S3;
918: x = bar( );
922: S4;
926: if (cont) goto L;
903: Speculative Thread:
908: L: S1;
912: fork(L);
916: S2;
920: foo(x);
924: S3;
928: x = bar( );
930: S4;
932: if (cont) goto L;

FIG. 9

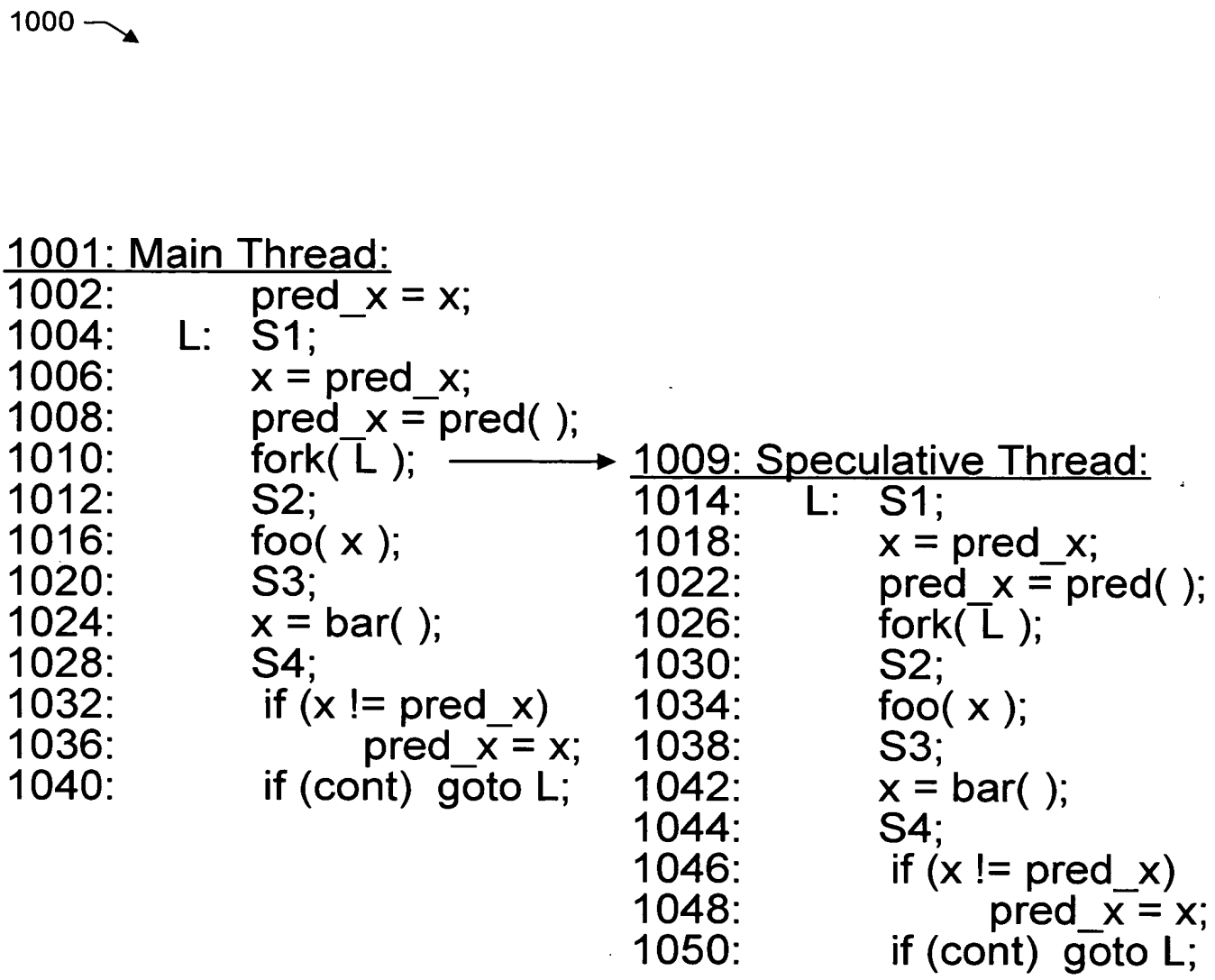

1000
1001: Main Thread:
1002: pred_x = x;
1004: L: S1;
1006: x = pred_x;
1008: pred_x = pred( );
1010: fork(L);
1012: S2;
1016: foo(x);
1020: S3;
1024: x = bar( );
1028: S4;
1032: if (x != pred_x)
1036: pred_x = x;
1040: if (cont) goto L;
1009: Speculative Thread:
1014: L: S1;
1018: x = pred_x;
1022: pred_x = pred( );
1026: fork(L);
1030: S2;
1034: foo(x);
1038: S3;
1042: x = bar( );
1044: S4;
1046: if (x != pred_x)
1048: pred_x = x;
1050: if (cont) goto L;

FIG. 10